The Claims

1. (Original) A system comprising:

a memory to store a plurality of triangle definition structures corresponding to a plurality of triangles, wherein each of the plurality of triangle definition structures includes a set of three vertices for the triangle, and wherein each of the plurality of triangle definition structures further includes a set of three edges, wherein each of the three edges corresponds to one of the three vertices, and wherein each of the three edges is an identification of the next edge that is encountered when performing a traversal in a particular direction about the corresponding vertex; and

a plurality of modules, coupled to be able to operate on the plurality of triangle definition structures, where each of the plurality of modules includes one or more instructions, and wherein each of the plurality of modules, when executed, manipulates various aspects of one or more of the plurality of triangle definition structures;

wherein one of the plurality of modules comprises a remove triangle process, wherein the remove triangle process removes a triangle definition structure from the plurality of triangle definition structures.

2. (Original) A system as recited in claim 1, wherein the particular direction comprises a counter-clockwise direction.

3. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises an add triangle module that adds a new triangle definition structure to the plurality of triangle definition structures.

4. (Original) A system as recited in claim 3, wherein the add triangle module receives, as an input, a set of three vertices from which to create the new triangle definition structure.

5. (Canceled).

6. (Currently amended) A system as recited in claim 1[[5]], wherein the remove triangle process further modifies one or more other definition structures of the plurality of triangle definition structures to account for the triangle definition structure being removed.

7. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a rotate operator that receives an edge of a triangle as an input, and returns a next edge in the triangle in the counterclockwise direction.

8. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises an inverse rotate operator that receives an edge of a triangle as an input, and returns the next edge in the triangle in the clockwise direction.

9. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a next edge operator that receives an edge of a triangle as an input, and returns the next edge from the triangle when rotating in a counterclockwise direction about a vertex of the triangle that is the origin of the input edge.

10. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a previous edge operator that receives an edge of a triangle as an input, and returns the next edge from the triangle when rotating in a clockwise direction about a vertex of the triangle that is the origin of the input edge.

11. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a same operator that receives an edge of a triangle as an input, and returns an edge of another triangle that has the same vertices as the input edge.

12. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises an origin operator that receives an edge of a triangle as an input, and returns a vertex that is an origin of the edge.

13. (Currently amended) A system ~~as recited in claim 1,~~ comprising:

a memory to store a plurality of triangle definition structures corresponding to a plurality of triangles, wherein each of the plurality of triangle definition structures includes a set of three vertices for the triangle, and wherein each of the plurality of triangle definition structures further includes a set of three edges, wherein each of the three edges corresponds to one of the three vertices, and wherein each of the three edges is an identification of the next edge that is encountered when performing a traversal in a particular direction about the corresponding vertex; and

a plurality of modules, coupled to be able to operate on the plurality of triangle definition structures, where each of the plurality of modules includes one or more instructions, and wherein each of the plurality of modules, when executed, manipulates various aspects of one or more of the plurality of triangle definition structures;

wherein one of the plurality of modules comprises a destination operator that receives an edge of a triangle as an input, and returns a vertex that is a destination of the edge.

14. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a right operator that receives an edge of a triangle as an input, wherein the input edge has an origin vertex, and wherein the right operator returns another vertex of an edge that shares the origin vertex with the input edge.

15. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a left operator that receives an edge of a triangle as an input, and returns a vertex that is a destination of the next edge from the triangle when rotating in a counterclockwise direction about a vertex of the triangle that is the origin of the input edge.

16. (Currently amended) A system ~~as recited in claim 1,~~ comprising:

a memory to store a plurality of triangle definition structures corresponding to a plurality of triangles, wherein each of the plurality of triangle definition structures includes a set of three vertices for the triangle, and wherein each of the plurality of triangle definition structures further includes a set of three edges, wherein each of the three edges corresponds to one of the three vertices, and wherein each of the three edges is an identification of the next edge that is encountered when performing a traversal in a particular direction about the corresponding vertex; and

a plurality of modules, coupled to be able to operate on the plurality of triangle definition structures, where each of the plurality of modules includes one or more instructions, and wherein each of the plurality of modules, when executed, manipulates various aspects of one or more of the plurality of triangle definition structures;

wherein one of the plurality of modules comprises a get representative edge operator that receives a vertex of a triangle as an input, and returns a representative edge of the vertex.

17. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a set representative edge operator that receives both an edge of a triangle and a vertex of the triangle as inputs, and sets the input edge to be the representative edge of the input vertex.

18. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a set origin operator that receives both an edge of a triangle and a vertex of the triangle as inputs, and sets a vertex of the triangle corresponding to an origin of the edge to be the input vertex.

19. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a set next edge operator that receives both a first edge and a second edge as inputs, and sets one of the edges in the triangle definition structure for the triangle that the first edge is part of to be the second edge.

20. (Currently amended) A system ~~as recited in claim 1,~~ comprising:

a memory to store a plurality of triangle definition structures corresponding to a plurality of triangles, wherein each of the plurality of triangle definition structures includes a set of three vertices for the triangle, and wherein each of the plurality of triangle definition structures further includes a set of three edges, wherein each of the three edges corresponds to one of the three vertices, and wherein each of the three edges is an identification of the next edge that is encountered when performing a traversal in a particular direction about the corresponding vertex; and

a plurality of modules, coupled to be able to operate on the plurality of triangle definition structures, where each of the plurality of modules includes one or more instructions, and wherein each of the plurality of modules, when executed, manipulates various aspects of one or more of the plurality of triangle definition structures;

wherein one of the plurality of modules comprises a make edge operator that receives both a first vertex and a second vertex as inputs, and generates two triangle definition structures, wherein the two triangle definition structures have adjacent edges between the first vertex and the second vertex.

21. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a make edge operator that receives both a first vertex and a second vertex as inputs, and generates two triangle definition structures to be added to the plurality of triangle definition structures, wherein each of the two triangle definition structures identifies, as the set of three vertices of the corresponding triangles, the input first vertex, the input second vertex, and a third vertex that is a boundary vertex.

22. (Original) A system as recited in claim 21, wherein each of the two triangle definition structures further identifies, as the set of three edges for the corresponding triangles, edges of the other triangle.

23. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a splice operator that receives both a first edge and a second edge as inputs, alters the set of edges of the triangle definition structure of the triangle including the first edge, and alters the set of edges of the triangle definition structure of the triangle including the second edge.

24. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a splice operator that receives both a first edge and a second edge as inputs, and wherein the splice operator:

identifies a third edge as a next edge, from a first triangle including the first edge, when rotating in a counterclockwise direction about a vertex of the first triangle that is the origin of the first edge;

identifies a fourth edge as a next edge, from a second triangle including the second edge, when rotating in a counterclockwise direction about a vertex of the second triangle that is the origin of the second edge;

identifies a fifth edge as a next edge in the first triangle in the counterclockwise direction;

identifies a sixth edge as a next edge in the second triangle in the counterclockwise direction;

identifies a seventh edge as a next edge in the first triangle in the clockwise direction;

identifies an eighth edge as a next edge in the second triangle in the clockwise direction;

sets, as a next edge that would be encountered when rotating in a counterclockwise direction from the first edge about the origin of the first edge, the fourth edge;

sets, as a next edge that would be encountered when rotating in a counterclockwise direction from the second edge about the origin of the second edge, the third edge;

sets, as a next edge that would be encountered when rotating in a counterclockwise direction from the fifth edge about the origin of the fifth edge, the eighth edge; and

sets, as a next edge that would be encountered when rotating in a counterclockwise direction from the sixth edge about the origin of the sixth edge, the seventh edge.

25. (Currently amended) A system as recited in claim 24, wherein ~~one~~ another of the plurality of modules comprises a swap operator that receives a particular edge of a triangle as an input, and wherein the swap operator:

identifies a ninth edge as a next edge, from the input particular edge, when rotating in a counterclockwise direction about the origin of the input particular edge;

identifies a tenth edge as a next edge in the input triangle in the counterclockwise direction from the input particular edge;

identifies a destination vertex of the ninth edge;

identifies a destination vertex of the tenth edge;

sets an origin of the ninth edge to be the destination vertex of the tenth edge;

sets an origin of the tenth edge to be the destination vertex of the ninth edge;

invokes the splice operator and uses, as inputs to the splice operator, the ninth edge and the input particular edge;

identifies an eleventh edge having the same vertices as the tenth edge; and

again invokes the splice operator and uses, as inputs to the splice operator, the ninth edge and the eleventh edge.

26. (Currently amended) A system as recited in claim 1, wherein ~~one~~ another of the plurality of modules comprises a swap operator that receives an edge of a triangle as an input, and returns an opposite diagonal of a quadrilateral corresponding to the input edge.

27-41. (Cancelled).

42. (Original) A method for adding a triangle to a triangular mesh, the triangle having three vertices and three edges, the method comprising:

receiving an indication of the three vertices of the triangle;

checking whether any of the three edges of the triangle already exist in the triangular mesh;

if any of the three edges of the triangle already exist in the triangular mesh, then checking connectivity of the mesh, determining if any connectivity of the triangular mesh needs to be changed, and changing connectivity of the triangular mesh as necessary to accommodate the triangle;

creating additional edges for the triangle as necessary until two edges of the triangle exist; and

connecting free ends of the two edges with an additional edge to create the triangle.

43. (Original) A method as recited in claim 42, further comprising setting a representative edge for each vertex of the triangle as necessary.

44. (Original) A method as recited in claim 42, further comprising setting a representative edge for each vertex of the triangle.

45. (Original) A method as recited in claim 42, wherein changing connectivity of the triangular mesh as necessary to accommodate the triangle comprises altering an edge identifier corresponding to another triangle of the triangular mesh to identify an edge of the triangle as a next edge that is encountered when performing a traversal in a particular direction about a vertex of the other triangle.

46. (Original) One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 42.

47. (Original) A method for removing a triangle from a triangular mesh, wherein each triangle in the triangular mesh includes an identification of three edges, and wherein each of the three edges is an identification of the next edge that is encountered when performing a traversal about a vertex corresponding to the edge, the method comprising:

receiving an identifier of the triangle to be removed;

updating a representative edge of each vertex of the triangle so that the representative edge is not an edge of the triangle; and

for each edge of the triangle, removing the edge and changing the connectivity of other triangles in the triangular mesh so that any other triangle having an identification of a next edge that is an edge of the triangle being removed has the identification changed to another edge of the triangular mesh.

48. (Original) A method as recited in claim 47, wherein the updating comprises, for each vertex:

identifying an edge of the triangle, wherein the edge has an origin at the vertex;

identifying, based on the edge, a previous edge that is the next edge from the edge when rotating in a clockwise direction about the vertex;

identifying a right vertex based on the previous edge, wherein the right vertex is another vertex of another edge that also has the origin as one of its vertices;

checking whether the right vertex is a boundary vertex; and

if the right vertex is not a boundary vertex, then setting the representative edge for the vertex to be the previous edge.

49. (Original) A method as recited in claim 48, wherein for each vertex if the right vertex is a boundary vertex, then the updating further comprises:

identifying, based on the previous edge, another previous edge that is the next edge from the previous edge when rotating in a clockwise direction about the vertex;

identifying, based on the edge, a subsequent edge that is the next edge from the edge when rotating in a counter-clockwise direction about the vertex;

checking whether the other previous edge and the subsequent edge are the same edges;

if the other previous edge and the subsequent edge are the same edges, then setting the representative edge for the vertex to be empty; and

if the other previous edge and the subsequent edge are not the same edges, then identifying, based on the other previous edge, a third previous edge that is the next edge from the other previous edge when rotating in a clockwise direction about the vertex, and setting the representative edge for the vertex to be the third previous edge.

50. (Original) A method as recited in claim 47, further comprising selectively invoking a make edge operator, wherein the make edge operator receives both a first vertex and a second vertex as inputs, and generates two triangles, wherein each of the two triangles identifies, as a set of three vertices of the corresponding triangles, the input first vertex, the input second vertex, and a third vertex that is a boundary vertex, and wherein each of the two triangles further identifies, as a set of three edges for the corresponding triangles, edges of the other triangle.

51. (Original) One or more computer-readable media having stored thereon a plurality of instructions to manage a triangular mesh, wherein the plurality of instructions, when executed by one or more processors of a computer, causes the one or more processors to perform the following acts:

implementing a make edge operator that receives both a first vertex and a second vertex as inputs, and generates two triangles, wherein the two triangles have adjacent edges between the first vertex and the second vertex;

implementing a splice operator that receives both a first edge and a second edge as inputs, alters connectivity of a triangle including the first edge, and alters connectivity of a triangle including the second edge; and

implementing a swap operator that receives a particular edge of a triangle as an input, and returns an opposite diagonal of a quadrilateral corresponding to the input edge.

52. (Original) One or more computer-readable media as recited in claim 51, wherein the instructions that cause the one or more processors to implement the make edge operator include instructions that cause the one or more processors to perform the following acts:

receiving both the first vertex and the second vertex as inputs; and

generating the two triangles, wherein each of the two triangles identifies, as a set of three vertices of the corresponding triangle, the input first vertex, the input second vertex, and a third vertex that is a boundary vertex, and wherein each of the two triangles further identifies, as a set of three edges for the corresponding triangle, edges of the other triangle.

53. (Original) One or more computer-readable media as recited in claim 51, wherein the instructions that cause the one or more processors to implement the splice operator include instructions that cause the one or more processors to perform the following acts:

receiving both the first edge and the second edge as inputs;

identifying a third edge as a next edge encountered, from the first edge, when rotating in a counterclockwise direction about a vertex that is the origin of the first edge;

identifying a fourth edge as a next edge encountered, from the second edge, when rotating in a counterclockwise direction about a vertex that is the origin of the second edge;

identifying a fifth edge as a next edge, in the triangle including the first edge, in the counterclockwise direction;

identifying a sixth edge as a next edge, in the triangle including the second edge, in the counterclockwise direction;

identifying a seventh edge as a next edge, in the triangle including the first edge, in the clockwise direction;

identifying an eighth edge as a next edge, in the triangle including the second edge, in the clockwise direction;

setting, as a next edge that would be encountered when rotating in a counterclockwise direction from the first edge about the origin of the first edge, the fourth edge;

setting, as a next edge that would be encountered when rotating in a counterclockwise direction from the second edge about the origin of the second edge, the third edge;

setting, as a next edge that would be encountered when rotating in a counterclockwise direction from the fifth edge about the origin of the fifth edge, the eighth edge; and

setting, as a next edge that would be encountered when rotating in a counterclockwise direction from the sixth edge about the origin of the sixth edge, the seventh edge.

54. (Original) One or more computer-readable media as recited in claim 53, wherein the instructions that cause the one or more processors to implement the swap operator include instructions that cause the one or more processors to perform the following acts:

receiving the particular edge of the triangle as an input;

identifying a ninth edge as a next edge, from the input particular edge, when rotating in a counterclockwise direction about the origin of the input particular edge;

identifying a tenth edge as a next edge in the input triangle in the counterclockwise direction from the input particular edge;

identifying a destination vertex of the ninth edge;

identifying a destination vertex of the tenth edge;

setting an origin of the ninth edge to be the destination vertex of the tenth edge;

setting an origin of the tenth edge to be the destination vertex of the ninth edge;

invoking the splice operator and using, as inputs to the splice operator, the ninth edge and the input particular edge;

identifying an eleventh edge having the same vertices as the tenth edge; and

again invoking the splice operator and using, as inputs to the splice operator, the ninth edge and the eleventh edge.

55. (Original) One or more computer-readable media as recited in claim 51, wherein the plurality of instructions further cause the one or more processors to perform the following acts:

selectively invoking the make edge operator, the splice operator, and the swap operator to add triangles to the triangular mesh; and

selectively invoking the make edge operator, the splice operator, and the swap operator to remove triangles from the triangular mesh.